# Free University of Bozen-Bolzano – Faculty of Computer Science
# Master of Science in Computer Science
# Theory of Computing – A.Y. 2008/2009
## Final exam – 30/1/2009 – Part 1

***Time: 90 minutes***

This is a closed book exam: the only resources allowed are blank paper, pens, and your head. Explain your reasoning. Write clearly, in the sense of logic, language and legibility. The clarity of your explanations affects your grade. Write your name and ID on every solution sheet. Good luck!

**Problem 1.1** [6 points] You must give a brief explanation of your answer to receive full credit.

(a) Decide whether the following statement is TRUE or FALSE: The complement of a recursive-enumerable language is recursively-enumerable.

(b) Let $M_2$ be a 2-tape (deterministic) TM, and let $M_1$ be the result of converting $M_2$ into a (1-tape deterministic) TM. Sketch using two to three sentences the main idea underlying the conversion. How are the running times of $M_1$ and $M_2$ related to each other?

(c) Decide whether the following statement is TRUE or FALSE: For all languages $L_1$, $L_2$, and $L_3$, if there exist a reduction from $L_1$ to $L_3$ and a reduction from $L_2$ to $L_3$, then there exists a reduction from $L_1$ to $L_2$.

**Problem 1.2** [6 points] Construct a TM $M$ that accepts the language $L = \{n\#w \mid n$ is a number represented in binary with the least significant digit on the *right*, and $w \in \{\mathtt{a},\mathtt{b},\mathtt{c}\}^*$ with $|w|_\mathtt{a} + |w|_\mathtt{b} = n\}$, where $|w|_x$ denotes the number of occurrences of $x$ in $w$.

E.g.: `10#accbc` $\in L$, `0#` $\in L$, `10#accbcb` $\notin L$ `10#ccac` $\notin L$.

Show the sequence of IDs of $M$ on the input strings "`10#acbc`" and "`10#cb`".

**Problem 1.3** [6 points] The *extraction* $L_1 \ominus L_2$ of two languages $L_1$ and $L_2$ is defined as:

$$L_1 \ominus L_2 = \{vw \mid vw_2w \in L_1, \text{ for some } w_2 \in L_2\}$$

Show that the class of recursively enumerable languages is closed under the *extraction* operation, i.e., that if $L_1$ and $L_2$ are recursively enumerable, then so is $L_1 \ominus L_2$.

[*Hint*: Show how to construct, from two (deterministic) TMs $M_1$ accepting $L_1$ and $M_2$ accepting $L_2$, a (possibly multi-tape) non-deterministic TM $N$ accepting $L_1 \ominus L_2$. You need not detail completely the construction of $N$, but you should provide sufficient information to make clear that you would be able to detail the construction, if you were given enough time to do so.]

**Problem 1.4** [6 points]

(a) Let $f$ and $g$ be primitive recursive functions. Show that the following predicate $p$ is primitive recursive:

$$p(x) = \begin{cases} 1 & \text{if } f(i) > g(j), \text{ for all } 1 \leq i \leq x \text{ and } 1 \leq j \leq x \\ 0 & \text{otherwise} \end{cases}$$

(b) Show that the following function $f$ is primitive recursive:

$$f(x) = \begin{cases} 2 & \text{if } x = 0 \text{ or } x = 1 \\ 3 \cdot f(x-1) \cdot f(x-2) & \text{if } x \geq 2 \end{cases}$$

**Problem 1.5** [6 points]

(a) Let $f$ be a total number-theoretic function with $n+1$ variables. Provide the definition of the $(n+1)$-variable function $gn_f$ such that $gn_f(\vec{x}, y)$ encodes the values of $f(\vec{x}, i)$ for $1 \leq i \leq y$.

(b) Let $g$ and $h$ be total number-theoretic functions, respectively with $n$ and $n+2$ variables. Define the $(n+1)$-variable function $f$ obtained from $g$ and $h$ by *course-of-values recursion*.